Regression Testing

Milos Gligoric

gliga@illinois.edu

cs427
11/04/2014

NSF CCF 10-12759
FDA: Software Responsible for 24% Of All Medical Device Failures

Source: iHS Global Insight, Reuters, FT research * Middle East, Central and South America, Africa

Toyotaa - total number of recalls = 8.6mio worldwide
Regression Testing

- Widely used in industry
- Executes tests for each new revision
- Checks if changes broke something
Regression Testing Cost

- Apache Ant: ~5min
- Guava Libraries: ~15min
- Jetty: ~45min
- Continuum: ~45min
- Apache Camel: ~45min
- Hadoop: ~4h
- ~17h
Regression Test Selection (RTS)

• Optimizes regression testing
• Analyzes changes to a codebase
• Runs only tests whose behavior may be affected

original revision  modified revision

changes

all affected tests => safe test selection
Regression Test Selection (Example)

changes to $p, q$

original revision

<table>
<thead>
<tr>
<th>m</th>
<th>p</th>
<th>q</th>
</tr>
</thead>
<tbody>
<tr>
<td>t1</td>
<td>⭐</td>
<td></td>
</tr>
<tr>
<td>t2</td>
<td></td>
<td>⭐</td>
</tr>
<tr>
<td>t3</td>
<td></td>
<td>⭐</td>
</tr>
<tr>
<td>t4</td>
<td>⭐</td>
<td>⭐</td>
</tr>
</tbody>
</table>

modified revision

<table>
<thead>
<tr>
<th>m</th>
<th>p</th>
<th>q</th>
</tr>
</thead>
<tbody>
<tr>
<td>t1</td>
<td>⭐️</td>
<td></td>
</tr>
<tr>
<td>t2</td>
<td></td>
<td>⭐</td>
</tr>
<tr>
<td>t3</td>
<td></td>
<td>⭐</td>
</tr>
<tr>
<td>t4</td>
<td>⭐</td>
<td>⭐</td>
</tr>
</tbody>
</table>

$\text{rts(}\text{original},\text{modified})$
Current Practice

• Few systems used in practice: Google™ TAP
  – Mapping of tests based on dependencies across projects
  – Not applicable to day-to-day work within single project

• No widely adoptable automated RTS tool after ~30 years of research

• Developers’ options:
  – RetestAll (expensive) or manual RTS (imprecise)
Real Time Data

- Data was captured using a record-and-replay tool, CodingTracker.
- Data had info not just on commits, but also on test sessions (runs of 1 or more tests).
- **Live data allowed us to study manual RTS**
Study Setup

• 14 developers working on 17 projects
• 3 months study
• 918 hours of development, 5757 test sessions, 264,562 executed tests
• 5 professional programmers, 9 UIUC students
Manual vs. Automated RTS

- Precision and safety
- ~70% of the time, Manual RTS > Auto RTS
  - Potentially wasting time
- ~30% of the time, Manual RTS < Auto RTS
  - Potentially missing faults

THE WORLD NEEDS REGRESSION TEST SELECTION

- Tests are taking a lot of time
- Developers are doing a poor job with Manual RTS
RTS: Execution vs. Overhead

changes to $p, q$

original revision

modified revision

$\text{rts(original,modified)}$

Analysis | Execution of Selected Coverage

Analysis Coverage $\Rightarrow$ Execution of non-Selected
Fine-grained Mappings

- Mapping from test to various code elements
  - edge in CFG, method, basic block, statement, ....

Implementation Complexity
Integration Complexity
Linear Software Histories

• Traditional test selection
  – Two revisions of code at a time

• Easy to extend to a linear sequence
  – Centralized version control systems (CVS, SVN, etc.)
Distributed Software Histories

- Distributed version control systems (e.g., Git)
- Complex graphs created by branching, merging, etc.

How to extend rts for distributed software histories?
Frequent Branching and Merging

- Analyzed 27 open-source projects on GitHub
- 30% of revisions are merges
Test Selection for Merge Command

<table>
<thead>
<tr>
<th></th>
<th>m</th>
<th>p</th>
<th>q</th>
</tr>
</thead>
<tbody>
<tr>
<td>t1</td>
<td>✨</td>
<td></td>
<td></td>
</tr>
<tr>
<td>t2</td>
<td></td>
<td>✨</td>
<td></td>
</tr>
<tr>
<td>t3</td>
<td></td>
<td></td>
<td>✨</td>
</tr>
<tr>
<td>t4</td>
<td>✨</td>
<td>✨</td>
<td></td>
</tr>
</tbody>
</table>

\[ S_{\text{merge}}^1(h) = rts(\text{imd}(h), h) \]

Pro: Runs traditional test selection only once (i.e., fast)

Con: There may be many changes between imd(h) and h => many tests selected to run (i.e., slow)
Merge Command (Option $S^0$)

<table>
<thead>
<tr>
<th>m</th>
<th>p</th>
<th>q</th>
</tr>
</thead>
<tbody>
<tr>
<td>t1</td>
<td>✹</td>
<td></td>
</tr>
<tr>
<td>t2</td>
<td></td>
<td>✹</td>
</tr>
<tr>
<td>t3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>t4</td>
<td>✹</td>
<td>✹</td>
</tr>
</tbody>
</table>

If a test is affected on multiple branches, changes from different branches together may lead to different results.

**Pro**: Does not run traditional test selection, but uses history results.

**Con**: Selects more tests than $S^i$ (e.g., new tests in one of the branches)

$$S^0_{merge}(h) = S_{aff}(h) \cup \left( A(h) \setminus \bigcap_{p \in pred(h)} A(p) \right)$$

$$S_{aff}(h) = \bigcup_{p,p' \in pred(h), p \neq p', d = \text{dom}(p,p')} \left( \bigcup_{n \in d \leq p \setminus \{d\}} S_{sel}(n) \right) \cap \left( \bigcup_{n \in d \leq p \setminus \{d\}} S_{sel}(n) \right)$$